

# Programação Orientada por Objetos 2020/2021

Licenciatura em Eng<sup>a</sup>. Informática

Boats & Docks – 1<sup>a</sup> Fase

Turma: 2	Data/Horário Lab.: 2 <sup>a</sup> Feira 08h30	Docente: Luís Miguel Claudino Cassaca
----------	--------------------------------------------------	---------------------------------------

Nº aluno	Nome	Endereço de Correio Eletrónico
200221060	Henrique Leote	200221060@estudantes.ips.pt

## Índice

1. Introdução e Apresentação do Jogo .....	3
2. Análise do Enunciado.....	5
3. Cartas CRC .....	11
4. Diagrama de Classes .....	12
5. Conclusões .....	13

## 1. Introdução e Apresentação do Jogo

Como projeto final da unidade curricular Programação Orientada por Objetos (POO), foi-nos proposto o desenvolvimento de um jogo de tabuleiro chamado “Boats & Docks”.

“Boats & Docks” é um jogo de tabuleiro tradicional, jogado individualmente, que consiste em atracar barcos nos portos/docas seguindo as restrições implementadas. O jogo tem como objetivo identificar as diferentes casas no tabuleiro, de acordo com determinadas regras, de forma a ser obtida a maior pontuação possível. O jogo possui um conjunto de casas cujo estado pode variar de entre os seguintes tipos: U – Desconhecido; W – Água, B – Barco e D – Porto. O objetivo principal consiste no jogador identificar em que estado se encontra cada posição. As posições correspondentes a “Água” podem ser alteradas para o estado “Barco” quando adjacentes a um porto/barco. As restrições são que cada barco deverá atracar num porto e todos os portos terão um e só um barco, sendo que as restantes casas constituirão o tipo “Água”, podendo o barco possuir três tipos de tamanhos: “pequeno”, “médio” ou “grande”. Cada “Barco” deverá estar numa das quatro casas adjacentes ao porto, não podendo estar nem tocar em barcos, tanto na horizontal como na vertical.

Quando o programa em si é executado, para já em ambiente de consola, é pedido ao utilizador para inserir o seu *nickname* (alunha) e, só após que o mesmo seja inserido de acordo com os critérios, pode escolher a dificuldade em que pretende jogar (Fácil, Médio ou Difícil). Aquando gerado e apresentado o tabuleiro ao utilizador, com um tamanho definido automaticamente mediante a dificuldade (Fácil – 5x5, Médio – 7x7, Difícil – 10x10) e com os campos aleatoriamente definidos (portos já definidos e as remanescentes posições desconhecidas), o utilizador pode começar a jogar, “transformando” posições desconhecidas em Água e/ou Barcos. Caso falhe, esse mesmo registo será guardado e adicionado à lista de erros desse mesmo jogador, penalizando a sua pontuação final. O jogo termina quando o tabuleiro estiver completo ou o jogador solicite a validação do mesmo, sendo que os campos restantes do tipo “desconhecido” alterarão o seu estado para “água”, sendo consequentemente exibida a pontuação final de jogo referente ao jogador em questão ( $\text{Pontuação final} = \text{pontuação base} - \text{penalizações} + \text{bonificações}$ ).

A pontuação é calculada tendo em conta a dificuldade do tabuleiro:

- Fácil – 50 pontos e 30 segundos base;
- Médio – 100 pontos e 90 segundos base;
- Difícil – 150 pontos e 150 segundos base.

Após efetuado o cálculo acima referido, o jogador pode ganhar ou perder pontos segundos os seguintes critérios:

- Pontos ganhos:
  - ✓ Falha na casa validada (água): -2 pontos por cada casa falhada;
  - ✓ Falha na casa validada (barco): -5 pontos por cada casa falhada;
  - ✓ Mais do que uma tentativa de validação: -10 por cada validação extra;
  - ✓ Tempo de jogo para além do tempo base: -1 ponto por cada segundo extra.
- Bonificações:
  - ✓ Apenas uma validação: +20 pontos;
  - ✓ Tempo de jogo inferior ao tempo base: +1 pontos por cada segundo não utilizado.

Nesta primeira fase de desenvolvimento do projeto foi realizada a análise do enunciado, obtendo todos os substantivos e verbos de maneira a formar as classes do programa. Foram também criadas cartas CRC (Class Responsibility Card) dessas mesmas classes. O tópico consequente abordará o reconhecimento do enunciado, através da análise dos substantivos anteriormente mencionados. Será igualmente abordada a criação de cartas CRC e o respetivo diagrama de classes.

Foram também implementados testes das classes de maneira a testar o programa e construção.

Este programa foi desenvolvido com recurso ao IDE (ambiente de desenvolvimento) NetBeans, pertencente à empresa Oracle.

## 2. Análise do Enunciado

Como referido na introdução, uma das etapas deste projeto consiste na análise do enunciado, realizada de forma a podermos classificar as potenciais classes, atributos e funcionalidades inerentes ao programa a ser desenvolvido.

De modo a realizarmos esta tarefa, procedemos à análise do enunciado referente ao projeto “Boats & Docks”, facultado pelo docente através plataforma de controlo de versões GitHub.

### 2.1. Seleção dos substantivos e verbos presentes no enunciado

#### 2.1.1. O Jogo

O objetivo deste jogo passa por identificar todas as casas do tabuleiro de jogo, de forma a obter a maior pontuação possível, por nível. Quando o jogo inicia, apresenta todas as casas do tipo porto, cabe ao jogador determinar quais as casas do tipo água e barco. Existem três dimensões de barcos diferentes, sendo estes pequenos, médios e grandes. O jogo termina quando cada porto tiver um barco atracado, seguindo as respetivas regras. Apenas quando colocados todos os barcos, o jogador pode validar o tabuleiro. Quando validado, todas as casas desconhecidas passam automaticamente ao estado água. Se o tabuleiro estiver correto, o jogo termina e é calculada a pontuação obtida. Caso contrário, o jogador é informado do número de erros existentes no tabuleiro. As casas erradas no momento da validação do tabuleiro de jogo, são declaradas como falhadas. Uma casa, uma vez falhada já não voltará a alterar o seu estado. Se a casa falhada for do tipo barco, as restantes casas que compõem o barco (caso existam), devem ser marcadas igualmente como falhadas. No final do jogo, é contabilizado o número total de casas falhadas e esse valor reflete uma penalização na pontuação do nível.

## 2.1.2. Casas de Jogo

Neste **jogo** existem duas tipologias de **casa** de **jogo**:

- **Casas** estáticas (inalteráveis)
  - ✓ **Porto** – identifica um **porto** marítimo, adjacente ao qual irá ser **atracado** um **barco**. Em cada **nível** são **dadas** a conhecer no **início do jogo** e não podem ser **alteradas**.
- **Casas** interativas (alteráveis) – o **jogador**, ao **clicar** nesta **casa**, **altera** o seu **tipo** entre a seguinte lista
  - **Desconhecida** – no **início do jogo**, todas as **casas** do **tabuleiro** que não são do tipo “**porto**”, são do tipo “**desconhecida**”. Esta representa uma **casa neutra**, cujo tipo ainda não foi identificado pelo **jogador**. Uma **casa** identificada como “**desconhecida**”, não contará para efeitos de **validação** de um **tabuleiro** de **jogo**. Um **jogador**, ao **clicar** numa **casa** deste tipo, irá **alterar** a sua **tipologia** para o tipo “**água**”.
  - **Água** – Esta **casa** representa um setor com **água** no **tabuleiro** de **jogo**. Esta **casa** fornece ao **jogador** dois **pontos** quando corretamente **colocada**. Um **jogador**, ao **clicar** numa **casa** deste tipo, **altera** a sua **tipologia** para o tipo “**barco**”.
  - **Barco** – Esta **casa** representa um **barco**, tem de ser **colocada** sempre adjacente a uma **casa** do tipo “**porto**”, segundo as **regras** do **jogo**. Esta **casa** fornece ao **jogador** cinco **pontos** quando corretamente **colocada**. Ao **clicar** nesta **casa**, volta ao tipo “**desconhecida**”. Podem existir **barcos** de três tamanhos distintos:
    - **Barco Pequeno** – composto por apenas uma **casa**, **totaliza** cinco **pontos** quando corretamente colocada.
    - **Barco Médio** – composto por duas **casas**, **totaliza** dez **pontos** quando corretamente colocadas.
    - **Barco Grande** – composto por três **casas**, **totaliza** quinze **pontos** quando corretamente colocadas.

Independentemente do tamanho do **barco**, este tem de estar **atracado** a um **porto**. Isto é, adjacente, quer vertical, quer horizontalmente. No entanto, este nunca pode ser **colocado** perpendicularmente ao respetivo **porto**. Esta **regra** não se aplica aos **barcos pequenos**, com apenas uma **casa**.

Após **dispor** todas as **casas** no **tabuleiro** de **jogo**, é possível **validar** a correta **colocação** das mesmas. Uma **casa** indevidamente **identificada**, deve **ser marcada** como **falhada**. Ainda que possa ser **alterada**, de forma que o **jogador** possa **completar** o **jogo**, a **marcação** da **falha** deve permanecer inalterável para efeitos de **pontuação**. Relembrar que, quando a **casa falhada** é do **tipo barco**, as restantes **casas** que o compõem serão igualmente **marcadas** como **falhadas**. Assim, quando **falhada** uma **casa** do tipo **barco**, este **deixa de contar** para bonificação na **pontuação final**.

### 2.1.3. Casas de Jogo

- Jogo;
- Casa;
- Tabuleiro;
- Pontuação Final;
- Nível;
- Porto;
- Jogador;
- Água;
- Barco;
- Pequeno (barco);
- Médio (barco);
- Grande (barco);
- Regras;
- Casas desconhecidas;
- Erros existentes ou casas erradas (tabuleiro);
- Falhas (casa);
- Estado (casa);
- Final do jogo;
- Valor (casas falhadas);
- Penalização;
- Início do jogo;
- Tipo/Tipologia (casa);
- Desconhecida (casa);
- Neutra (casa);
- Pontos (pontuação);
- Regra.

Consideraram-se substantivas as palavras que julgámos serem as mais significativas no desenvolvimento do jogo.

### 2.1.4. Verbos

Na análise dos verbos foram retiradas do enunciado frases que possam representar funcionalidades e/ou atributos.

- Quando o jogo inicia, apresenta todas as casas do tipo porto;
- No final do jogo, o jogador é informado se foi atingido uma nova pontuação máxima;
- A contagem do tempo inicia no momento da colocação da primeira casa de jogo;
- No fim do jogo juntamente com a pontuação obtida, é apresentada a sua melhor pontuação já alcançada até ao momento;
- Cabe ao jogador determinar quais as casas do tipo água e barco;
- Uma casa, uma vez falhada, já não voltará a alterar o seu estado;
- Ainda que possa ser alterada, de forma que o jogador possa completar o jogo, a marcação da falha deve permanecer inalterável para efeitos de pontuação;
- O jogador, ao clicar nesta casa, altera o seu tipo entre a seguinte lista:

Desconhecido ->Água -> Barco -> Desconhecido;

- Apenas quando colocados todos os barcos, o jogador pode validar o tabuleiro;
- Quando validado, todas as casas desconhecidas passam automaticamente ao estado água;
- Barco – Esta casa representa um barco, tem de ser colocada sempre adjacente a uma casa do tipo porto;
- Após dispor todas as casas no tabuleiro de jogo, é possível validar a correta colocação das casas;
- Se o tabuleiro estiver correto, o jogo termina e é calculada a pontuação obtida. Caso contrário, o jogador é informado do número de erros existentes no tabuleiro;
- As casas erradas no momento da validação do tabuleiro de jogo, são declaradas como falhadas;
- Os barcos nunca podem ser colocados perpendicularmente ao respetivo porto. Esta regra não se aplica aos barcos pequenos, com apenas uma casa;
- O jogo está terminando apenas quando todo o tabuleiro estiver corretamente preenchido;
- No final do jogo, é contabilizado o número total de casas falhadas e esse valor reflete uma penalização na pontuação do nível;
- A casa “Desconhecido” não contará para efeitos de validação de um tabuleiro de jogo;
- O jogador é identificado por um nome (nickname), introduzido pelo próprio jogador, que deverá ser único;
- Cada jogador deve manter um histórico de pontuações obtidas, para todos os jogos efetuados pelo próprio, com registo de data e hora do jogo;
- Relembrar que, quando a casa falhada é do tipo barco, as restantes casas que o compõem serão igualmente marcadas como falhadas. Assim, quando falhada uma casa do tipo barco, este deixa de contar para bonificação na pontuação final;
- Cada validação corresponde a uma tentativa, sendo que mais do que uma tentativa por nível implica uma penalização de 10 pontos;
- Um jogo completo sem qualquer falha, isto é, validado à primeira tentativa, incorre de uma bonificação de 20 pontos;
- O tempo total de jogo será comparado ao tempo de base definido para o nível em questão. Caso o tempo de jogo ultrapasse o tempo base, constitui uma penalização. Por outro lado, quando o tempo de jogo for inferior ao tempo base, constitui uma bonificação;



- Cálculo da pontuação de um nível: Pontuação Final = pontuação base + bonificações + penalizações;
- Deve ser possível aceder à pontuação máxima obtida pelo jogador, em cada um dos níveis;
- Deve ainda, ser apresentado um ranking, onde apresenta todos os jogadores, ordenados de forma decrescente, com base na pontuação acumulada de cada;
- No caso de obter uma pontuação máxima, esta será registada como a pontuação atual, obtida pelo jogador, para o nível em questão;
- Então, cada barco irá atracar num porto, e todos os portos terão um e só um barco;
- Os números à volta do tabuleiro identificam o número de barcos a colocar, na linha (horizontal) ou coluna (vertical) correspondente;
- Cada barco deve estar numa das quatro casas adjacentes ao porto em questão;
- Os barcos não se podem tocar, nem mesmo diagonalmente;
- As regras de validação do tabuleiro criado seguem as mesmas regras da validação do tabuleiro em momento de jogo, isto é, após a criação de um novo nível é necessário verificar se este não contém erros;
- Para além do tabuleiro de jogo, o nível deverá também, conter informação relativa ao identificador do nível (nome pelo qual deverá ser identificado), autor (quem criou) e a respetiva data de criação;
- A cada nível é associada uma pontuação base, definido pelo grau de dificuldade do mesmo;
- O utilizador poderá criar novos níveis. Para tal, deve determinar o grau de dificuldade do nível a ser criado. Mediante o grau de dificuldade, é determinado o número total de portos possíveis de colocar no tabuleiro: Fácil (5), Médio (10) e difícil (17).

#### Verbos:

- Identificar;
- Iniciar;
- Apresentar;
- Determinar;
- Terminar;
- Tiver (estar);
- Atracado (atracar);
- Colocado (colocar);
- Validar;
- Validado;
- Passam (passar);
- Calcular;
- Informar;
- Declarar;
- Voltar a alterar;
- Marcar;
- Contabilizar;
- Reflete (refletir);
- Dadas (dar);
- Alteradas (alterar);
- Clicar (selecionar);
- Altera (alterar);
- São (ser);
- Validação (validar);

- Colocada (colocar);
- Soma no total (totalizar);
- Atracado (atracar);
- Colocado (colocar);
- Dispor;
- Validar;
- Colocação (colocar);
- Identificada (Identificar);
- Ser marcada (marcar);
- Alterada (alterar);
- Completar;
- Marcação (marcar);
- Marcadas (marcar);
- Deixar de contar;

### 3. Cartas CRC

Jogo ( <i>Game</i> )	Colaboradores
<ul style="list-style-type: none"> <li>• Inicia o jogo;</li> <li>• Valida o tabuleiro;</li> <li>• Preenche as casa do tabuleiro;</li> </ul>	<ul style="list-style-type: none"> <li>• Jogador (<i>Player</i>);</li> <li>• Tabuleiro (<i>Board</i>);</li> <li>• Pontuação (<i>Score</i>);</li> <li>• Regras (<i>Rules</i>);</li> </ul>

Posição ( <i>Position</i> )	Colaboradores
<ul style="list-style-type: none"> <li>• Define a posição da casa no tabuleiro;</li> </ul>	

Jogador ( <i>Player</i> )	Colaboradores
<ul style="list-style-type: none"> <li>• Cria o jogador do jogo com uma pontuação mediante do tabuleiro;</li> </ul>	<ul style="list-style-type: none"> <li>• Jogo (<i>Game</i>);</li> <li>• Pontuação (<i>Score</i>);</li> </ul>

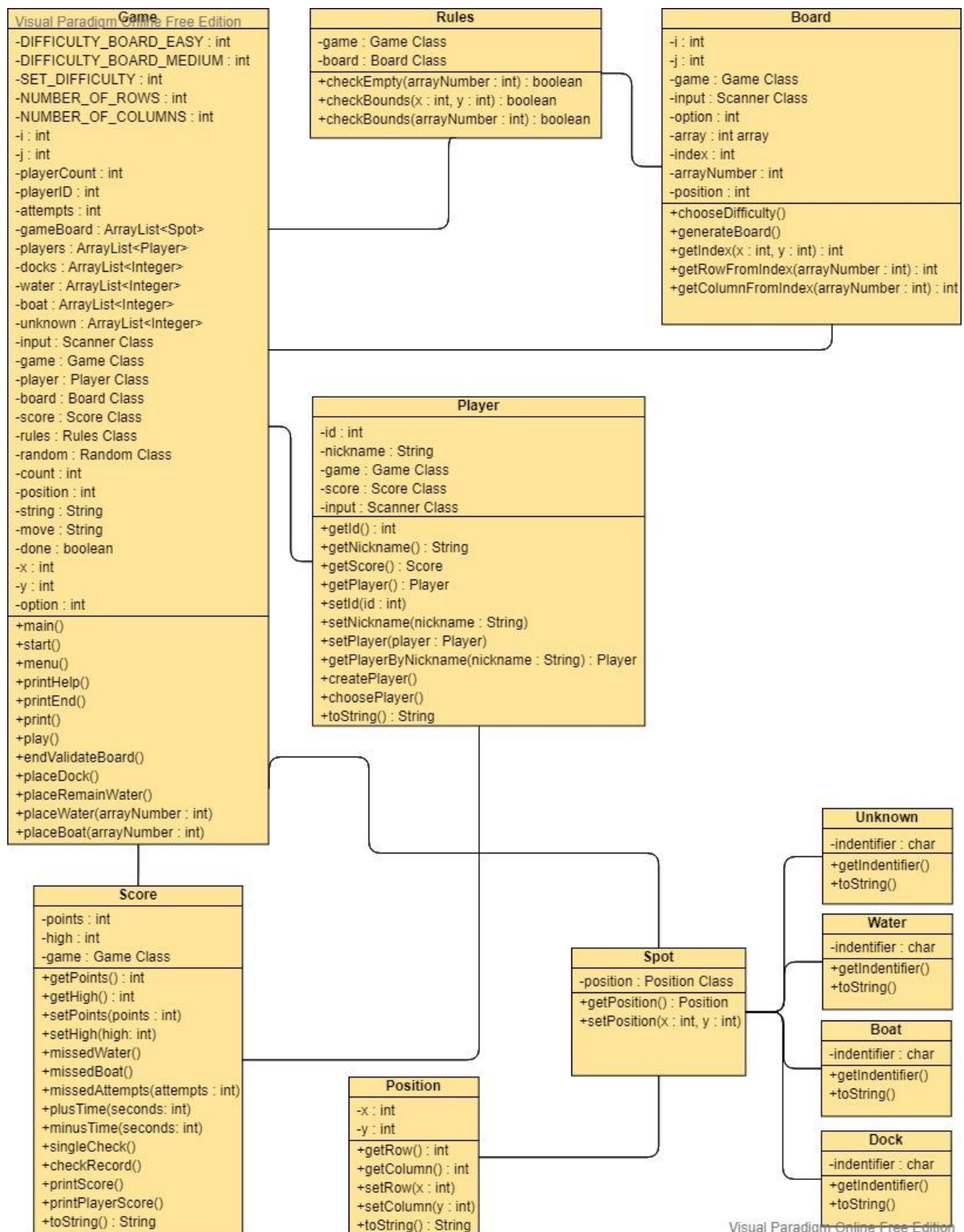
Pontuação ( <i>Score</i> )	Colaboradores
<ul style="list-style-type: none"> <li>• Define a pontuação do jogador;</li> </ul>	<ul style="list-style-type: none"> <li>• Jogo (<i>Game</i>);</li> </ul>

Tabuleiro ( <i>Board</i> )	Colaboradores
<ul style="list-style-type: none"> <li>• Escolhe a dificuldade do tabuleiro;</li> <li>• Gera o tabuleiro de jogo;</li> </ul>	<ul style="list-style-type: none"> <li>• Jogo (<i>Game</i>);</li> </ul>

Regras ( <i>Rules</i> )	Colaboradores
<ul style="list-style-type: none"> <li>• Verifica posições no tabuleiro;</li> </ul>	<ul style="list-style-type: none"> <li>• Tabuleiro (<i>Board</i>);</li> <li>• Jogo (<i>Game</i>);</li> </ul>

Casa ( <i>Spot</i> )	Colaboradores
<ul style="list-style-type: none"> <li>• Define o tipo de casa</li> </ul>	

## 4. Diagrama de Classes



Visual Paradigm Online Free Edition

## 5. Conclusões

Com a realização desta segunda fase deste projeto, foi consolidado o conhecimento em relação às heranças, polimorfismo, coleções e exceções e o quão importante são e facilitam na programação orientada a objetos.